

# Python! Comparative Crime Rate Alert.

Contributed by Kevin Bell  
03, Mar. 2011  
Last Updated 11, Mar. 2011

In order to keep a finger on the pulse of the city, I wrote a little code to do a historical comparison of the crime rate for 10 different crime types per police beat. For each beat and crime type, I compare the last 7 days crime frequency to the last 90 days and paint the police beat red if an increase has occurred. WITH OUR TEST DATA, it appears that spring is in the air, and the burglars are coming out of their winter hibernation! The code below writes a rate into my police beats feature class every night. After a little tweaking, this will be used to determine where officers may want to focus. Let me know if you have any trouble getting it set up. There is a bit of vbScripting in the labels, and the mxd has to be set up right to get this look.

By turning on the other layers, like car prowls, etc, you'll see those labels come up too.

```
# -----
# pd_comparitiveCrimeRates.py
# Purpose: for each police beat and each general offense crime type,
#         compare the last 7 days to the last 90 days and flag beats
#         that have and increase in any crime type.
#
# Author: Kevin Bell, Salt Lake City IT

# Revision:
#         Aaron Marvel, SLC PD, (equation tuning)
# Date: 20110215
#-----

from datetime import date, timedelta
import pprint
import arcpy

arcpy.env.overwriteOutput = True
arcpy.env.workspace = r'C:\GISLocal\CrimeData\community\20110209_hotspotsFor2010\Hotspots2010.gdb'
city = r'C:\GISLocal\shapes\bm_cityboundary.shp'
beatFC = r'C:\GISLocal\shapes\slcpdBeats.shp'
fc = r'C:\GISLocal\CrimeData\community\Community_Council.gdb\Community_Council_NAD1927'
masterDict = {}

arcpy.MakeFeatureLayer_management(beatFC, 'beatfclyr')
arcpy.MakeFeatureLayer_management(fc, 'fclyr')
arcpy.SelectLayerByLocation_management('fclyr', 'INTERSECT', city)

def updateBeatFC(beat, k, n):
    rows = arcpy.UpdateCursor(r'C:\GISLocal\CrimeData\20110210_crimeSpikeDetection\default.gdb\beats')
    for r in rows:
        if r.BEAT_ID == beat:
            if k == 'Arson':
                r.Arson = n
            elif k == 'Assault Aggravated':
                r.AggAssault = n
            elif k == 'Burglary':
                r.Burglary == n
            elif k == 'Larc-Car Prowl':
                r.CarProwl = n
            elif k == 'Larc-Car Strip':
                r.CarStrip = n
            elif k == 'Robbery':
                r.Robbery == n
            elif k == 'Sex Assault':
```

```

    r.SexAssault = n
elif k == 'Stolen Vehicle':
    r.StolenVehicle = n
elif k == 'Weapon-Drive By':
    r.WeaponDriveBy = n
else:
    pass
rows.updateRow(r)
del r, rows

```

```

def makeLayer(k, v):
    """make a feature layer per crime type"""
    arcpy.MakeFeatureLayer_management('fclyr', k + '_lyr', v)
    print 'made lyr ' + k

def getPastCrimes(lyr, days, offset):
    """get the last n Days of data for a crime type layer"""
    #-- Offset the crimes to not include the current day for the 7 day, and to start counting 90 days back from the end of the
    7 days
    d = date.today() - timedelta(days + offset)
    off = date.today() - timedelta(offset)
    print "report_date" >= date ' + "" + str(d) + "" + ' AND "report_date" < date ' + "" + str(off) + ""
    arcpy.SelectLayerByAttribute_management(lyr, 'NEW_SELECTION', "report_date" >= date ' + "" + str(d) + "" + ' AND
"report_date" < date ' + "" + str(off) + ""
    i=0
    for r in arcpy.SearchCursor(lyr):
        i += 1
    print str(i)
    return float(i)

def getComparitiveCrimeRatePerBeat(beatLyr, beat):
    for k, v in offenseDict.iteritems():
        makeLayer(k, v)
        try:
            #-- GET THE AVERAGE FOR 7 DAYS
            crime_seven = float(getPastCrimes(k + '_lyr', 8.0, 0.0))/7
            #-- GET THE AVERAGE FOR 90 DAYS
            crime_ninety = float(getPastCrimes(k + '_lyr', 90.0, 8.0))/90
            #-- Check if the 90 day average is 0
            if (crime_ninety == 0):
                comparitiveRate = 0
            else :
                #-- GET THE % change in crime from the 7 day average and the 90 day average
                #-- Equation -- (7 day average - 90 day average) / 90 day average -- END Equation ----
                comparitiveRate = (crime_seven - crime_ninety) / crime_ninety

            print str(comparitiveRate)
            updateBeatFC(beat, k, comparitiveRate)

        except:
            print 'failed to get comparitive rate for beat ' + str(beat)
            pass

#-----
print "Create Offense Dictionary"
offenseDict = {}
for r in arcpy.SearchCursor(fc):
    offenseDict[r.type] = r"type" = ' + "" + r.type + ""
pprint.pprint(offenseDict)

print "Create Beats Array"
beats = []

```

```
for r in arcpy.SearchCursor(beatFC):  
    beats.append(r.BEAT_ID)
```

```
for beat in beats:
```

```
    print '----- ' + str(beat) + ' -----'  
    print 'beatfclyr', 'NEW_SELECTION', "'BEAT_ID" = ' + "" + str(beat) + ""  
    arcpy.SelectLayerByAttribute_management('beatfclyr', 'NEW_SELECTION', "'BEAT_ID" = ' + str(beat))  
    arcpy.SelectLayerByLocation_management('fclyr', 'INTERSECT', 'beatfclyr')  
    arcpy.CopyFeatures_management('fclyr', 'in_memory\\tempBeat')  
    getComparitiveCrimeRatePerBeat('in_memory\\tempBeat', beat)
```